Feature_counts

2024-01-21

```
setwd("/Users/tiknokeziah/Desktop/Keziah/School/Research Project/Data/Count Data")

library(ggplot2)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(limma)
library(edgeR)
```

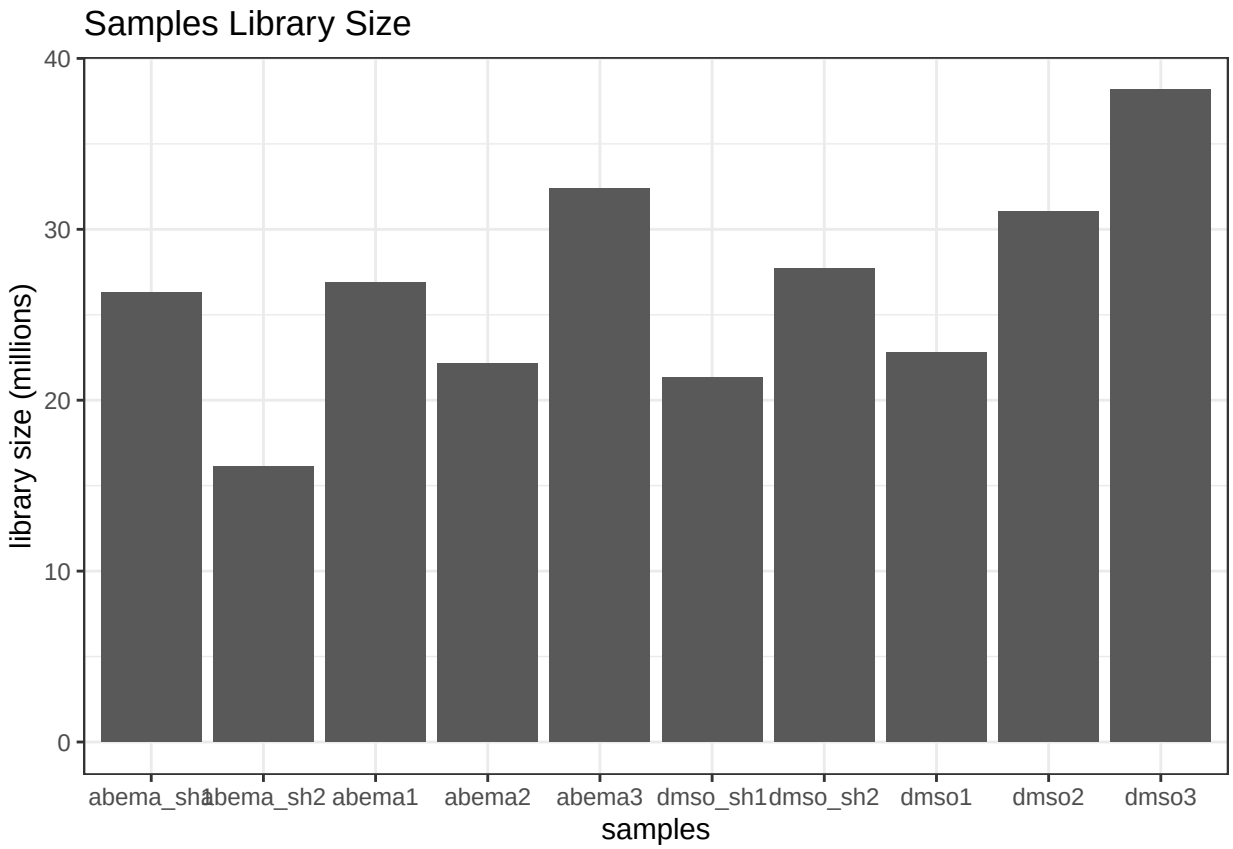
```
## Warning: package 'edgeR' was built under R version 4.3.2
```

```
fc <- readRDS("Feature_Counts.RData")

counts_data <- data.frame(colSums(fc$counts))
colnames(counts_data) <- "lib_size"
counts_data$gene_ID <- rownames(counts_data)
rownames(counts_data) <- NULL

counts_data <- counts_data %>% relocate(gene_ID)
counts_data$lib_size <- counts_data$lib_size/1e06
counts_data$type <- c("dmso1", "dmso2", "dmso3", "abema1", "abema2", "abema3", "dmso_sh1", "dmso_sh2",

ggplot(counts_data, aes(x = type, y=lib_size)) +
  geom_bar(stat="identity") +
  ggtitle("Samples Library Size") +
  xlab("samples") +
  ylab("library size (millions)") +
  theme_bw()
```



Filtering lowly expressed genes

```
# Obtain CPMs
myCPM <- cpm(fc$counts)
# Have a look at the output
head(myCPM)
```

```
##           SRR12576402 SRR12576403 SRR12576404 SRR12576405 SRR12576406
## 100287102   0.8772769  0.77371255   0.8911717  1.33749964  1.58118530
## 653635     24.8708001 28.56288826  29.0417126 20.73124449 19.42599088
## 102466751   0.0000000  0.09671407   0.1834765  0.03715277  0.04517672
## 100302278   0.0000000  0.00000000   0.0000000  0.00000000  0.00000000
## 645520      0.0000000  0.00000000   0.0000000  0.00000000  0.00000000
## 79501       0.0000000  0.00000000   0.0000000  0.00000000  0.00000000
##           SRR12576407 SRR12576408 SRR12576409 SRR12576410 SRR12576411
## 100287102  1.08153163  1.03215714  0.90227037   0.5319779    1.923495
## 653635    23.94820044 15.10702719 33.60054852  16.9092974   18.862656
## 102466751  0.09270271  0.00000000  0.03609081   0.1139953    0.000000
## 100302278  0.00000000  0.09383247  0.00000000   0.0000000    0.000000
## 645520     0.00000000  0.04691623  0.00000000   0.0000000    0.000000
## 79501      0.00000000  0.00000000  0.00000000   0.0000000    0.000000
```

```
# Which values in myCPM are greater than 0.5?
thresh <- myCPM > 0.5
# Logical matrix with TRUEs and FALSEs
head(thresh)
```

```
##           SRR12576402 SRR12576403 SRR12576404 SRR12576405 SRR12576406
## 100287102        TRUE        TRUE        TRUE        TRUE        TRUE
## 653635           TRUE        TRUE        TRUE        TRUE        TRUE
## 102466751       FALSE       FALSE       FALSE       FALSE       FALSE
## 100302278       FALSE       FALSE       FALSE       FALSE       FALSE
## 645520          FALSE       FALSE       FALSE       FALSE       FALSE
## 79501           FALSE       FALSE       FALSE       FALSE       FALSE
##           SRR12576407 SRR12576408 SRR12576409 SRR12576410 SRR12576411
## 100287102        TRUE        TRUE        TRUE        TRUE        TRUE
## 653635           TRUE        TRUE        TRUE        TRUE        TRUE
## 102466751       FALSE       FALSE       FALSE       FALSE       FALSE
## 100302278       FALSE       FALSE       FALSE       FALSE       FALSE
## 645520          FALSE       FALSE       FALSE       FALSE       FALSE
## 79501           FALSE       FALSE       FALSE       FALSE       FALSE
```

```
# Summary of how many TRUEs there are in each row
# 13046 genes that have TRUEs in all 12 samples
table(rowSums(thresh))
```

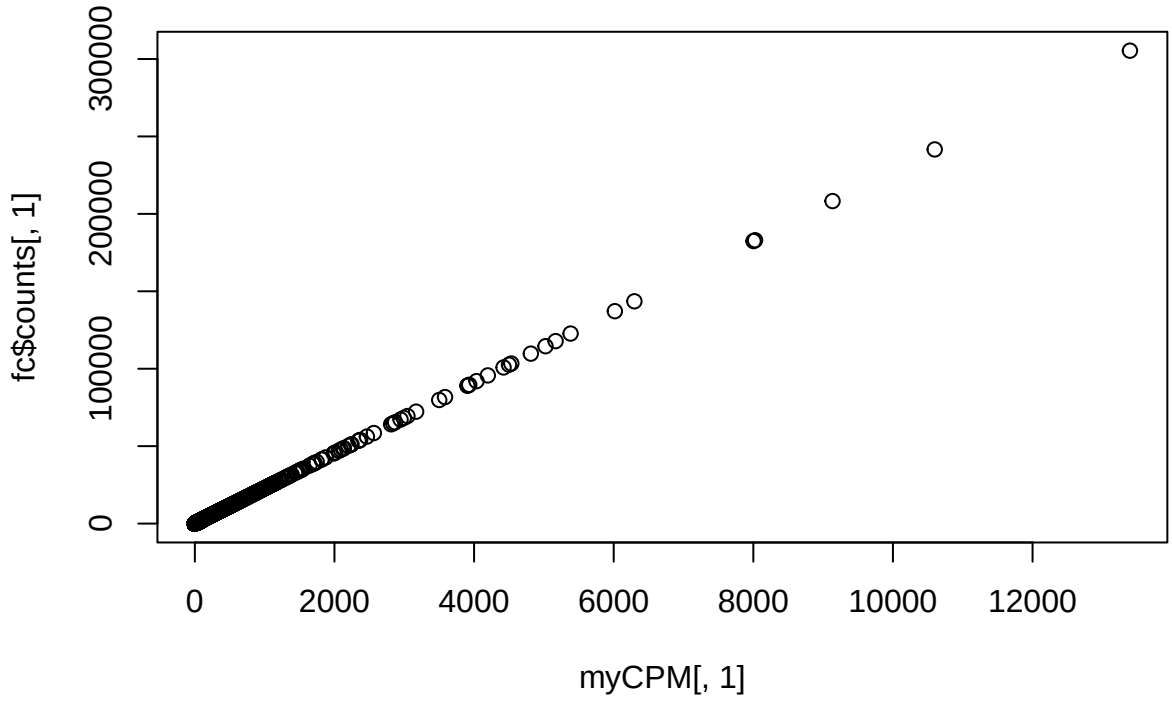
```
##
##     0     1     2     3     4     5     6     7     8     9    10
## 12648   723   278   270   216   213   197   247   216   341 13046
```

```
# Keep genes that have at least 2 TRUEs in each row of the thresh
keep <- rowSums(thresh) >= 2
summary(keep)
```

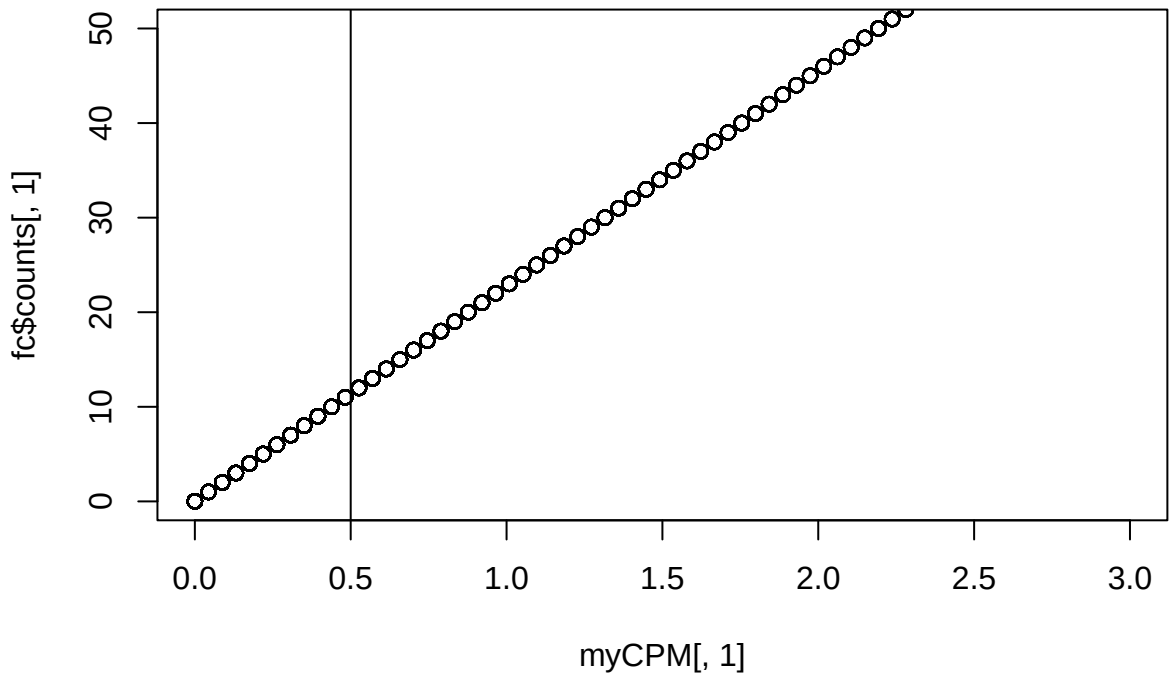
```
##    Mode   FALSE    TRUE
## logical   13371   15024
```

Library size and distribution plots

```
# Let's have a look and see whether our threshold of 0.5 does indeed correspond to a count of about 10-
# We will look at the first sample
plot(myCPM[,1], fc$counts[,1])
```



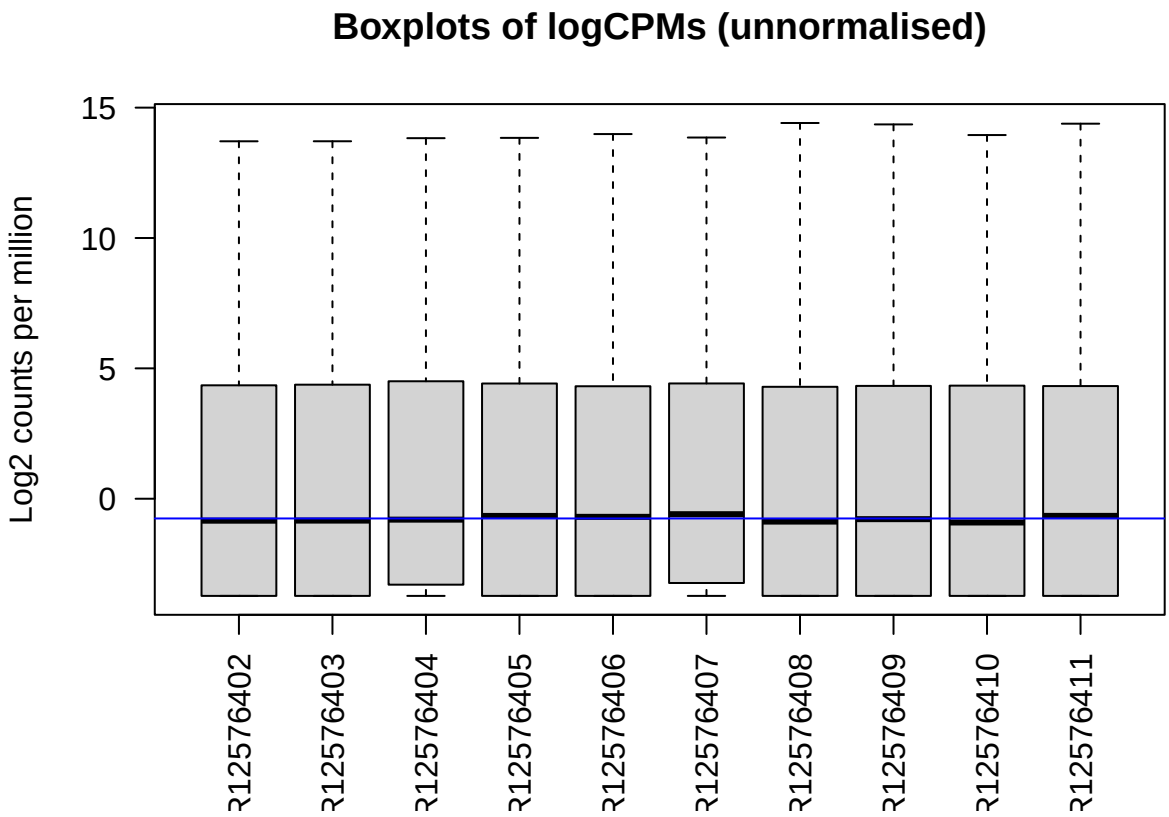

```
plot(myCPM[,1], fc$counts[,1], ylim=c(0,50),xlim=c(0,3))
abline(v=0.5)
```




```
# Get log2 counts per million
logcounts <- cpm(fc$counts, log=TRUE)

# Check distributions of samples using boxplots
## Change with ggplot
boxplot(logcounts, xlab="", ylab="Log2 counts per million", las=2)
```

```
abline(h=median(logcounts),col="blue")
title("Boxplots of logCPMs (unnormalised)")
```




```
plotMDS(fc$counts)
```